means [(620, 640)] operative on the non-branch instructions in each said basic block for processing said instructions, and

means [(620, 1548)] operative on said branch instruction in said basic block in response to the firing time information for completing the execution of said branch instruction during the same time [(IFT)] as said processing means is processing the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block [in order to speed up] thereby speeding the overall processing of said program by said system.

72. (Amended) A system for executing branches in single entry-single exit (SESE) basic blocks (BBs) in a plurality of programs utilized by a number of users, each basic block having a plurality of non-branch instructions and a branch instruction, said system comprising:

means [(160)] receptive of each said programs for determining the branch instruction within each said basic block of each of said programs, said determining means [being] further [capable of] adding firing time information [(IFT)] to said branch instructions,

means [(620, 640)] operative on the non-branch instructions in each said basic block of each said program for processing said programs, and

means [(620, 1548)] operative on said branch instructions in each said basic block in response to the firing time information for completing the execution of said branch instruction during the same time [(IFT)] as said processing means is processing the last to be executed non-branch instruction in said basic block for a given program so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block [in order to speed] thereby speeding up the overall processing all of said programs by said system.

74. (Amended) A system for executing branches in single entry-single exit (SESE) basic blocks (BBs) contained within a program, said basic block having a plurality of non-branch instructions and a branch instruction, said system comprising:

means [(620)] receptive of said program for determining the branch instruction within each said basic block of said program, said determining means [being] further [capable of] scheduling processing of said branch instruction,

means [(620, 640)] operative on the non-branch instructions in each said basic block for processing said instructions, and

means [(620, 1548)] operative on said branch instruction in said basic block for completing the execution of said scheduled branch instruction no later than during the same time as said processing means is processing the last to be executed non-branch instruction in said basic block so that the execution of said

branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block [in order to speed] thereby speeding up the overall processing of said program by said system.

75. (Amended) A system for executing branches in single entry-single exit (SESE) basic blocks (BBs) in a plurality of programs utilized by a number of users, said basic block having a plurality of non-branch instructions and a branch instruction, said system comprising:

means [(160)] receptive of each said programs for determining the branch instruction within each said basic block of each of said programs, said determining means [being] further [capable of] scheduling processing of said branch instructions,

means [(620, 640)] operative on the non-branch instructions in each said basic block of each said program for processing said programs, and

means [(620, 1548)] operative on said branch [instructions] instruction in each said basic block for completing the execution of said scheduled branch instruction no later than during the same time as said processing means is processing the last to be executed non-branch instruction in said basic block for a given program so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block whereby overall processing throughput of all said programs by said system is increased.

76. (Amended) A system for executing scheduled branches in single entry-single exit (SESE) basic blocks (BBs) contained within a program, each basic block having a plurality of non-branch instructions and a branch instruction, said system comprising:

means [(620)] receptive of said program for determining the branch instruction within each said basic block of said program said determining means [being] further [capable of] adding instruction firing time information to said scheduled branch instruction,

means [(620, 640)] operative on the [instruction] non-branch instructions in each said basic block for processing said non-branch instructions, and

means [(620, 1548)] operative on said branch instruction in said basic block in response to said time information, for completing the execution of said scheduled branch instruction during the same time as said processing means is processing the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block [in order to speed] thereby speeding up the overall processing of said program by said system.

77. (Amended) A system for executing scheduled branches in single entry-single exit (SESE) basic blocks (BBs) in a plurality of programs utilized by a number of users, each basic block having a plurality of non-branch instructions and a branch instruction, said system comprising:

means [(160)] receptive of each said programs for determining the branch instruction within each said basic block of each of said programs, said determining means [being] further [capable of] adding instruction firing time information to said scheduled branch instructions,

means [(620, 640)] operative on the non-branch instructions in each said basic block of each said program for processing said programs, and

means [(620, 1548)] operative on said branch instructions in each said basic block for completing the execution of said scheduled branch instruction during the same time as said processing means is processing the last to be executed non-branch instruction in said basic block for a given program so that the execution of said branch instruction occurs in parallel with the execution of said instructions in said basic block whereby overall processing throughput of all said programs by said system is increased.

78. (Amended) A method for executing branches in single entry-single exit (SESE) basic blocks (BBs) contained within a program, each said basic block having a plurality of non-branch

instructions and a branch instruction, said method comprising the steps of:

determining the branch instruction within each said basic block of said program,

adding information to said branch instruction,

processing said instructions in each said basic block, and

completing the execution of said branch instruction in said basic block, based upon said added information, no later than during the time duration of processing the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block [in order to speed] thereby speeding up the overall processing of said program.

79. (Amended) A method for parallel processing natural concurrencies in a program using a plurality of processor elements [(PEs)], said program having a plurality of single entry-single exit (SESE) basic blocks (BBs) with each of said basic blocks (BBs) having a stream of instructions including a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

determining the natural concurrencies within said instruction stream in each of said basic blocks (BBs) in said program,

adding intelligence to each instruction in each said basic block in response to the determination of said natural concurrencies, said added intelligence at least comprising an

instruction firing time [(IFT)] and a logical processor number [(LPN)] so that all processing resources required by any given instruction are allocated in advance of processing, and

processing the instructions having said added intelligence in said plurality of processor elements corresponding to the logical processor numbers, each of said plurality of processor elements receiving all instructions for that processor in the order of the instruction having earliest instruction firing times, the instruction having the earliest time being delivered first.

C1

Sub E3

80. (Amended) A method for executing branches in single entry-single exit (SESE) basic blocks (BBs) contained within a program, each basic block having a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

determining the branch instruction within each said basic block of said program,

scheduling processing of said branch instruction,

processing said instructions in each said basic block, and

completing the execution of said scheduled branch instruction no later than during the processing of the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block [in order to speed] thereby speeding up the overall processing of said program.

81. (Amended) A method for executing branches in single entry-single exit (SESE) basic blocks (BBs) in a plurality of programs utilized by a number of users, each basic block having a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

determining the branch instruction within each said basic block of each said programs,

scheduling processing of said branch instructions,

processing the instructions in each said basic block of each said program, and

completing the execution of said scheduled branch instruction no later than during the processing of the last to be executed non-branch instruction in said basic block for a given program so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block whereby overall processing throughput of all said programs is increased.

82. (Amended) A method of executing scheduled branches in single entry-single exit (SESE) basic blocks (BBs) contained within a program, each basic block having a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

determining the branch instruction within each said basic block of said program,

adding instruction firing time information to said branch instruction for scheduling said [scheduled] branch instruction,

processing said instructions in each said basic block, and

completing the execution of said scheduled branch instruction according to said firing time information no later than during the processing of the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block [in order to speed] thereby speeding up the overall processing of said program.

83. (Amended) A method for executing scheduled branches in single entry-single exit (SESE) basic blocks (BBs) in a plurality of programs utilized by a number of users, each basic block having a plurality of non-branch instructions and a branch instruction, said method comprising the steps of:

determining the branch instruction within each said basic block of each of said programs,

adding instruction firing time information to said scheduled branch instruction for scheduling processing of said branch instruction,

processing the instructions in each said basic block of said programs, and

completing the execution of said scheduled branch instruction according to said firing time information no later than during the processing of the last to be executed non-branch instruction in